

JEGYZŐKÖNYV

Adatbázis rendszerek MSc

2021. tavasz féléves feladat

Készítette: **Gőz Franciska Noémi**

Neptunkód: **JNHQE0**

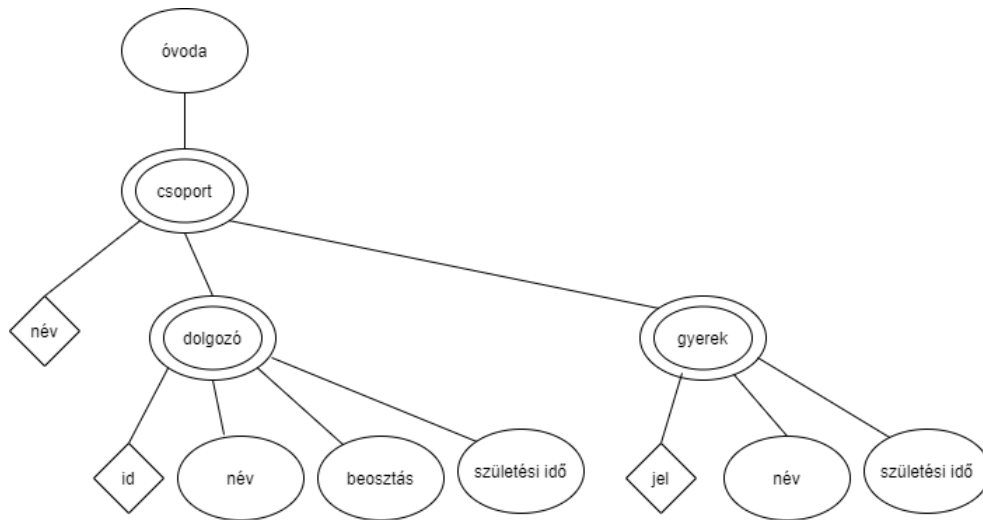
Adatbázis rendszerek MSc

Gyakorlat

2020.03.16.

1. feladat

Adott az alábbi logikai modell:



Hozzuk létre az XML dokumentumot.

Valósítsuk meg XQuery-vel a következő feladatokat:

- óvónők kilistázása abc sorrendben
- azonos csoportba járó gyerekek neveinek kilistázása
- legidősebb dadus nevének lekérdezése
- legfiatalabb óvónő nevének lekérdezése
- új gyerek felvitele
- gyerek adatainak módosítása
- gyerek törlése
- azonos jelű gyerekek neveinek kilistázása
- életkort kiszámító függvény készítése
- csoportok létszámának meghatározása

Megoldás

a)

```
ovoda.xml 10
1 xquery version "3.1";
2
3 for $o in doc("ovoda.xml")//dolgozo
4 where $o/beosztas="óvónő"
5 order by $o/nev
6 return
7   element {"óvónő"} {
8     text {$o}
9   }
```

/db/gyakorlat/ovoda//10

Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches

```
1 <óvónő>
    Bea
    óvónő
    1979
  </óvónő>
2 <óvónő>
    Enikő
    óvónő
    1968
  </óvónő>
3 <óvónő>
    Erika
    óvónő
    1978
  </óvónő>
4 <óvónő>
    Ildikó
    óvónő
    1988
  </óvónő>
5 <óvónő>
    Katalin
    óvónő
    1976
  </óvónő>
```

b)

```
ovoda.xml 10 20
1 xquery version "3.1";
2
3 for $cs in fn:doc("ovoda.xml")//csoport
4 where $cs/@nev = 'süni'
5 return
6   element {"sünicsoport"} {
7     text {$cs/gyerek/nev}
8   }
```

/db/gyakorlat/ovoda//20

Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index

```
1 <sünicsoport>Noémi Péter Bence Andrea Patrik Emma Ágnes</sünicsoport>
```

c)

```
ovoda.xml  10  20  30
1  xquery version "3.1";
2
3  for $d in fn:doc("ovoda.xml")//dolgozo
4  where $d/beosztas="dadus" and $d/szuletesiev=min(//szuletesiev)
5  return
6      <legidoesebbdadus>
7          { $d/nev }
8          { $d/szuletesiev }
9      </legidoesebbdadus>
```

↶ /db/gyakorlat/ovoda//3o

⏪ Adaptive Output ▾ ☒ Indent ☐ Live Preview ☒ Highlight Index Matches ⌕

```
1 <legidoesebbdadus>
  <nev>Márta</nev>
  <szuletesiev>1966</szuletesiev>
</legidoesebbdadus>
```

d)

```
ovoda.xml  10  20  30  40
1  xquery version "3.1";
2
3  for $d in fn:doc("ovoda.xml")//dolgozo
4  where $d/beosztas="óvónő" and $d/szuletesiev=max(//dolgozo/szuletesiev)
5  return
6      <legfiatalabbóvónő>
7          { $d/nev }
8          { $d/szuletesiev }
9      </legfiatalabbóvónő>
```

↶ /db/gyakorlat/ovoda//4o

⏪ Adaptive Output ▾ ☒ Indent ☐ Live Preview ☒ Highlight Index Matches ⌕

```
1 <legfiatalabbóvónő>
  <nev>Ildikó</nev>
  <szuletesiev>1988</szuletesiev>
</legfiatalabbóvónő>
```

e)

```
ovoda.xml  10  20  30  40  50
1  xquery version "3.1";
2
3  let $gy := fn:doc("ovoda.xml")//csoport[@nev="maci"]
4  return
5  update insert
6      <gyerek jel="traktor">
7          <nev>Benőke</nev>
8          <szuletesiev>2018</szuletesiev>
9      </gyerek>
10 into $gy
```

↻ /db/gyakorlat/ovoda//5o

⏪ Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches [+](#)

```

</gyerek>
<gyerek jel="traktor">
  <nev>Benőke</nev>
  <szuletesiev>2018</szuletesiev>
</gyerek>
```

f)

```
ovoda.xml  10  20  30  40  50  60
1  xquery version "3.1";
2
3  let $gy := fn:doc("ovoda.xml")//gyerek
4  return
5  update value //nev [.="Benőke"] with "Benedek"
```

```

</gyerek>
<gyerek jel="traktor">
  <nev>Benedek</nev>
  <szuletesiev>2018</szuletesiev>
</gyerek>
</csoport>
```

g)

```
ovoda.xml  10  20  30  40  50  60  70
1  xquery version "3.1";
2
3  let $gy := fn:doc("ovoda.xml")//gyerek[@jel = "traktor"]
4  return
5  update delete $gy
```

↻ /db/gyakorlat/ovoda//7o

⏪ Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches [+](#)

h)

```
ovoda.xml 10 20 30 40 50 60 70 80
1 xquery version "3.1";
2
3 for $n in fn:doc("ovoda.xml")//gyerek
4 where $n/@jel="napocska"
5 return
6   element {"eredmény"} {
7     attribute {"jel"} {$n/@jel},
8     text {$n/nev}
9   }

/db/gyakorlat/ovoda//8o
Adaptive Output Indent Live Preview Highlight Index Matches
1 <eredmény jel="napocska">Eszter</eredmény>
2 <eredmény jel="napocska">Noémi</eredmény>
3 <eredmény jel="napocska">Anna</eredmény>
```

i)

```
ovoda.xml 10 20 30 40 50 60 70 80 90
1 xquery version "3.1";
2
3 declare namespace fv="f1";
4 declare function fv:etelkor($s) as xs:integer{
5   let $e := fn:year-from-date(fn:current-date()) - $s cast as xs:integer
6   return $e
7 };
8
9 <eredmeny>
10 {
11   let $doc := fn:doc("ovoda.xml")
12   for $gy in $doc//gyerek
13   return
14     element {"gyerek"} {
15       attribute {"jel"} {$gy/@jel},
16       attribute {"név"} {$gy/nev},
17       text {fv:etelkor(number($gy/szuletesiev))}
18     }
19 }
20 </eredmeny>

/db/gyakorlat/ovoda//9o
Adaptive Output Indent Live Preview Highlight Index Matches
1 <eredmeny>
  <gyerek jel="napocska" név="Eszter">6</gyerek>
  <gyerek jel="autó" név="Gábor">6</gyerek>
  <gyerek jel="maci" név="Eliza">5</gyerek>
  <gyerek jel="szívecske" név="Lilla">4</gyerek>
  <gyerek jel="repülő" név="Soma">6</gyerek>
  <gyerek jel="alma" név="Kristóf">5</gyerek>
  <gyerek jel="méhecske" név="Maja">4</gyerek>
  <gyerek jel="baba" név="Anna">4</gyerek>
  <gyerek jel="körte" név="Lili">4</gyerek>
  <gyerek jel="napocska" név="Noémi">3</gyerek>
  <gyerek jel="autó" név="Péter">3</gyerek>
  <gyerek jel="maci" név="Bence">4</gyerek>
</eredmeny>
```

j)

```
ovoda.xml  10  20  30  40  50  60  70  80  90  100
1  xquery version "3.1";
2
3  for $cs in fn:doc("ovoda.xml")//csoport
4  return
5  element {"létszám"} {
6      attribute {"csoport"} {$cs/@nev},
7      text {count($cs/gyerek)}
8  }
```

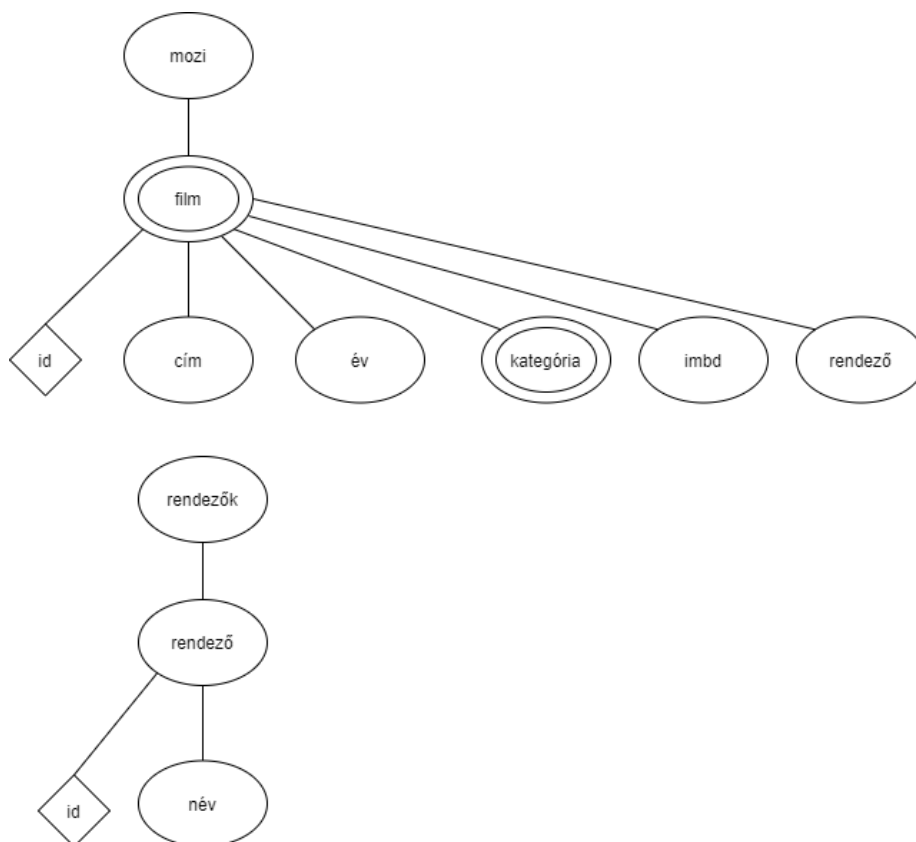
➡ /db/gyakorlat/ovoda//100

Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches

```
1 <létszám csoport="maci">9</létszám>
2 <létszám csoport="süni">7</létszám>
3 <létszám csoport="katicabogár">7</létszám>
```

2.feladat

Adottak az alábbi logikai modellek:



Hozzuk létre az XML dokumentumokat.

Valósítsuk meg XQuery-vel a következő feladatokat:

- a) 2000-es évek előtti filmek kilistázása
- b) átlagos játékidő kiszámolása
- c) filmek típusainak csoportosítása
- d) azonos kategóriában lévő filmek számának meghatározása
- e) a legjobb értékelésű film lekérdezése
- f) filmek rendezőikkel való összerendelése
- g) új film felvitele
- h) film adatainak módosítása
- i) film törlése
- j) filmek korának meghatározása

Megoldás

a)



The screenshot shows an XQuery editor with three tabs: 'mozi.xml', 'rendezok.xml', and '1m'. The query is as follows:


```
1 xquery version "3.1";
2
3 for $f in fn:doc("mozi.xml")//film
4 where $f/ev < 2000
5 order by $f/cim
6 return $f/cim
```

Below the query, the execution path is shown as '/db/gyakorlat/mozi/1m'. The output is displayed as:

```
1 <cim>Kincs, ami nincs</cim>
```

At the bottom, there are checkboxes for 'Adaptive Output' (checked), 'Indent' (checked), and 'Live Preview' (unchecked).

b)



The screenshot shows an XQuery editor with four tabs: 'mozi.xml', 'rendezok.xml', '1m', and '2m'. The query is as follows:

```
1 xquery version "3.1";
2
3 let $jatekido := fn:doc("mozi.xml")//jatekido
4 let $atlag:=avg($jatekido)
5
6 return
7     <eredmeny>
8         { $atlag }
9     </eredmeny>
```

Below the query, the execution path is shown as '/db/gyakorlat//mozi/2m'. The output is displayed as:

```
1 <eredmeny>98.16666666666667</eredmeny>
```

At the bottom, there are checkboxes for 'Adaptive Output' (checked), 'Indent' (checked), 'Live Preview' (unchecked), and 'High' (checked).

c)

The screenshot shows an XQuery editor interface. At the top, there are tabs for 'mozi.xml', 'rendezok.xml', and three memory usage indicators: '1m', '2m', and '3m'. The main editor area contains two XQuery snippets. The first snippet (lines 1-7) uses 'distinct-values' to find unique categories from 'mozi.xml'. The second snippet (lines 8-12) uses 'fn:doc' to read 'mozi.xml' and 'group by' to list categories. Below the editor, a toolbar shows a refresh icon, the path '/db/gyakorlat//mozi/3m', and settings for 'Adaptive Output', 'Indent', 'Live Preview', and 'Highlight Index Mat'. The results pane at the bottom displays a list of four categories: 'akció', 'animáció', 'dráma', and 'vígjáték'.

```
1 xquery version "3.1";
2
3 let $doc := fn:doc("mozi.xml")
4 for $kategoria in distinct-values($doc//film/kategoria)
5 order by $kategoria
6 return $kategoria
7
8 (:
9 for $kategoria in fn:doc("mozi.xml")//kategoria
10 group by $kategoria
11 return $kategoria
12 :)
```

/db/gyakorlat//mozi/3m

Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Mat

- 1 "akció"
- 2 "animáció"
- 3 "dráma"
- 4 "vígjáték"

d)

mozi.xmlrendezok.xml1m2m3m4m

```

1 xquery version "3.1";
2
3 let $doc := fn:doc("mozi.xml")
4 for $k in distinct-values($doc//film/kategoria)
5 return
6   element {"eredmény"} {
7     attribute {"kategória"} {$k},
8     text {count($doc//film[kategoria eq $k])}
9   }

```

/db/gyakorlat//mozi/4m

Adaptive Output
☒ Indent
☐ Live Preview
☒ Highlight In

```

1 <eredmény kategória="vígjáték">6</eredmény>
2 <eredmény kategória="animáció">3</eredmény>
3 <eredmény kategória="dráma">2</eredmény>
4 <eredmény kategória="akció">1</eredmény>

```

e)

mozi.xmlrendezok.xml1m2m3m4m5m

```

1 xquery version "3.1";
2
3 for $f in fn:doc("mozi.xml")//film
4 where $f/imbd = max(//imbd)
5 return $f

```

/db/gyakorlat//mozi/5m

Adaptive Output
☒ Indent
☐ Live Preview
☒ Highlight Index M

```

1 <film id="ab5">
  <cim>Életrevalók</cim>
  <ev>2011</ev>
  <kategoria>vígjáték</kategoria>
  <kategoria>dráma</kategoria>
  <jatekido>112</jatekido>
  <imbd>8.5</imbd>
  <rendezo>5</rendezo>
</film>

```

f)

```
mozi.xml    rendezok.xml    1m    2m    3m    4m    5m    6m
1  xquery version "3.1";
2
3  for $f in fn:doc("mozi.xml")//film
4  for $r in fn:doc("rendezok.xml")//rendezo
5  where $r/@id = $f/rendezo/text()
6  return
7  element {"rendezte"} {
8      attribute {"cim"} {$f/cim},
9      text {$r/nev/text()}
10 }
```

↩ /db/gyakorlat//mozi/6m

⏪ Adaptive Output ☒ Indent ☐ Live Preview ☒ Highlight Index Matches

```
1 <rendezte cim="Gru">Pierre Coffin</rendezte>
2 <rendezte cim="A bakancslista">Rob Reiner</rendezte>
3 <rendezte cim="Kincs, ami nincs">Sergio Corbucci</rendezte>
4 <rendezte cim="Shrek">Andrew Adamson</rendezte>
5 <rendezte cim="Életrevalók">Olivier Nakache</rendezte>
6 <rendezte cim="Shrek 2">Andrew Adamson</rendezte>
```

```

1 xquery version "3.1";
2
3 let $f := fn:doc("mozi.xml")/mozi
4 return
5   update insert
6     <film id="ab7">
7       <cim>A keresztapa</cim>
8       <ev>1972</ev>
9       <kategoria>krimi</kategoria>
10      <kategoria>dráma</kategoria>
11      <jatekido>175</jatekido>
12      <imbd>9.2</imbd>
13    </film>
14
15 into $f

```

```

<film id="ab7">
  <cim>A keresztapa</cim>
  <ev>1972</ev>
  <kategoria>krimi</kategoria>
  <kategoria>dráma</kategoria>
  <jatekido>175</jatekido>
  <imbd>9.2</imbd>
</film>

```

h)

The screenshot shows an XQuery editor with a file explorer at the top containing 'mozi.xml', 'rendezok.xml', and tabs '1m' through '8m'. The editor displays the following XQuery code:

```

1 xquery version "3.1";
2
3 let $gy := fn:doc("mozi.xml")//film
4 return
5   update value //cim [.="A keresztapa"] with "Keresztapa"

```

Below the code, the path '/db/gyakorlat//mozi/8m' is shown. At the bottom, there are controls for 'Adaptive Output', 'Indent' (checked), 'Live Preview' (unchecked), and 'Highlight Index Matches' (checked).

```

<film id="ab7">
  <cim>Keresztapa</cim>
  <ev>1972</ev>
  <kategoria>krimi</kategoria>
  <kategoria>dráma</kategoria>
  <jatekido>175</jatekido>
  <imbd>9.2</imbd>
</film>

```

i)

The screenshot shows an XQuery editor with a file explorer at the top containing 'mozi.xml', 'rendezok.xml', and tabs '1m' through '9m'. The editor displays the following XQuery code:

```

1 xquery version "3.1";
2
3 let $f := fn:doc("mozi.xml")//film[@id = "ab7"]
4
5 return
6   update delete $f

```

Below the code, the path '/db/gyakorlat//mozi/9m' is shown. At the bottom, there are controls for 'Adaptive Output', 'Indent' (checked), 'Live Preview' (unchecked), and 'Highlight Index Matches' (checked).

j)

mozi.xmlrendezok.xml1m2m3m4m5m6m7m8mmozi.xml9m10m

```
1 xquery version "3.1";
2
3 declare namespace fv="f1";
4 declare function fv:kor ($s) as xs:integer{
5   let $e := fn:year-from-date(fn:current-date()) - $s cast as xs:integer
6   return $e
7 };
8
9 <eredmeny>
10 {
11   let $doc := fn:doc("mozi.xml")
12   for $f in $doc//film
13   return
14     element {"film"} {
15       attribute {"cim"} {$f/cim},
16       attribute {"megjelenés"} {$f/ev},
17       text {fv:kor(number($f/ev))}
18     }
19 }
20 </eredmeny>
```

↶ /db/gyakorlat//mozi/10m

Adaptive Output

☒ Indent

☐ Live Preview

☒ Highlight Index Matches

⌕

```
1 <eredmeny>
  <film cím="Gru" megjelenés="2010">11</film>
  <film cím="A bakancslista" megjelenés="2007">14</film>
  <film cím="Kincs, ami nincs" megjelenés="1981">40</film>
  <film cím="Shrek" megjelenés="2001">20</film>
  <film cím="Életrevalók" megjelenés="2011">10</film>
  <film cím="Shrek 2" megjelenés="2004">17</film>
</eredmeny>
```

🟢

Query returned 1 item(s) in 0.016s